

**Remarks**

Claims 1-3, 5-14, 16-25, and 27-32 are pending in this application. Claims 4, 15, and 26 have been canceled without prejudice. Reconsideration of the application is respectfully requested in view of the following remarks.

The Office has asserted a rejection of 7-11, 18-22, 27, and 28-32 under 35 USC §102(e) as anticipated by Leiba, U.S. Patent No. 6,260,065 (“Leiba”) in an Office action dated September 25, 2006 (Office action). The Office has also asserted a rejection of claims 1-3, 5-6, 12-14, 16-17, 23-25, 27 and 28 under 35 USC §103(a) as obvious over Leiba in view of Willis, U.S. Patent No. 6,321,376 (“Willis”) in the same Office action. Applicants disagree with the rejections given and the Examiner’s characterizations. Applicants address these rejections below.

***1. With the goal of reaching a shared understanding of the disclosure of Leiba and Willis, the Applicants respectfully make the following observations.***

***A. Leiba***

Leiba describes a method for testing a server to ensure that it complies with a standard, such as an RFC standard. [Leiba, 3:58-61.] The server is pictured as a stand-alone device (FIG. 1, at 140) whose only contact with the tester is through test commands that are fed into the server, and responses to the test commands that the server passes back to the tester -- e.g., “providing configuration information and a sequence of test commands to the server application, receiving at least one response from the server application and comparing the at least one response with expected responses for performing compliance testing based on the response requirements.” [Leiba, 2: 4-8.]

The implementation of the server at a source code level is never discussed, nor is there any need to discuss or modify the source code of the server, as there is no procedure which incorporates anything but responses to test commands from the test engine 130.

As shown in figure 1, “client test data is processed by a test engine and then transmitted to a server. The server responses are then processed to create a test result.” [Leiba, 4:52-54.] The test data consists of two files: a test configuration file, which consists of server configuration information and test information, and a test input file, which consists of protocol commands and test elements. The software used to create the server itself under test is not used to generate test data; rather, the server is

treated as a black box consisting of inputs and outputs only, the outputs of which are used to determine if its response to the test data is correct.

The method of Leiba does not discuss the creation of the client test data. Rather it takes provided test commands, feeds them to the server being tested, and then analyzes the server responses. [Leiba, 4:1-7.] Rather than just allowing a right or a wrong answer for a given response from a server, a response can be of several types: mandatory, optional, forbidden, etc., (Leiba 2:45-47) allowing “server responses [to be expressed] in a flexible and general format....” [Leiba, 7:43.]

### ***B. Willis***

Willis describes a method to automatically “generat[e] and classify[] a set of good and bad language conformity tests from a manually generated, formal specification of the language being generated.” [Willis 2:56-58.] Once these language conformity tests are generated, they are then submitted to a “tool under test.” [Willis, 3:24, FIG. 1.] The software implementation of the tool under test is never referenced. Rather, the tool under test is treated as a black box which accepts test inputs and produces test outputs. [Willis, 3:17-28.]

## **2. *Patentability over Leiba in view of Willis***

The Office has also asserted a rejection of claims 1-3, 5-6, 12-14, 16-17, 23-25, 27 and 28 under 35 USC §103(a) as obvious over Leiba in view of Willis, U.S. Patent No. 6,321,376 (“Willis”). Applicants respectfully traverse. Leiba and Willis, taken separately or in combination, fail to teach or suggest at least one limitation of claims 4 claims 1-3, 5-6, 12-14, 16-17, 23-25, 27 and 28. Applicants respectfully submit the claims in their present form are allowable over the cited art.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. (MPEP § 2142.)

Motivations to combine or modify references must come from the references themselves or be within the body of knowledge in the art. (See MPEP § 2143.01.)

***A. The combination of Leiba-Willis is improper.***

The combination of Leiba and Willis proposed by the Examiner to reject claims 1, 3, 5, 7-9, 11, 13-16, 18-19, 23-26, and 30-31 is improper. As to Leiba, the Examiner states that “Leiba does not explicitly disclose compiling the software implementation source code from a first high-level language into an intermediate language; compiling the software specification from a second high-level language into the intermediate language; and producing the conformance-test enabled implementation in the intermediate language.” [Action, p. 5.] The Applicants agree. The Examiner argues, however, that Willis does so teach, and that:

it would have been obvious to one of ordinary skill in the art at the time of the invention was made to incorporate the method of compiling software implementation/specification from high level language into an intermediate language as taught by Willis in corresponding to the method of verifying software applications using conformance test as taught by Willis. The modification would be obvious because on of ordinary skill in the art would be motivated to compile software implementation/specification from high level language into an intermediate language to provide an efficient way of performing conformance testing as suggested by Willis. [Action, pp. 5-6.]

Applicants respectfully disagree.

***1. Impermissible hindsight***

It is impermissible to use the claims as an instruction manual or “template” to piece together the teachings of the prior art to render a claimed invention obvious. *See In re Fritch*, 972 F.2d 1260, 1266. “Virtually all [inventions] are combinations of old elements.” *In re Rouffet*, 47 USPQ.2d at 1457 (Fed Cir. 1998) . Thus, the invention “must be viewed not after the blueprint has been drawn by the inventor, but as it would have been perceived in the state of the art that existed at the time the invention was made without hindsight or knowledge of the invention.” *Sensonics, Inc. v. Aerisonic Corp.*, 38 USPQ.2d 1551, 1554 (Fed. Cir. 1996). “To draw on hindsight knowledge of the patented invention, when the prior art does not contain or suggest that knowledge, is to use the invention as a template for its own reconstruction -- an illogical and inappropriate process by which to determine patentability” *Id.*

Thus, the best argument against “hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or motivation to combine prior art references.” *In Re Dembicziak*, 50 USPQ.2d 1614, 1617 (Fed. Cir. 1999).

For the reasons stated above, the general statement in the Office Action to combine Leiba and Willis “to provide an efficient way of performing conformance testing” [Action at 6] is legally insufficient in that the office has failed to prove that they have not used hindsight to combine the teachings of Leiba and Willis.

Furthermore, obviousness can only be established by combining the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either explicitly or implicitly either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. MPEP 2143.01.

“In considering motivation in the obviousness analysis, the problem examined is not the specific problem solved by the invention but the general problem that confronted the inventor before the invention was made.” In re Rouffet, 47 USPQ.2d at 1457 (Fed. Cir. 1998).

The motivation itself can come from three sources: “the nature of the problem to be solved, the teachings of the prior art, and the knowledge of persons of ordinary skill in the art.” In re Rouffet, 47 USPQ.2d at 1458.

The Office Action does not provide a specific explanation why one of skill in the art at the time of the invention would choose the teachings from Leiba combined with the teachings from Willis to “compile implementation/specification from high level language into an intermediate language,” [Leiba, p. 6] over other potential solutions “to provide an efficient way of performing conformance testing”. [Id.] The office also does not provide a specific explanation as to why or how the method taught in Willis would make Leiba more efficient. Thus, the motivation is inadequate and applicants respectfully submit that the claim is in condition for allowance.

## *2. The modification changes the principle of operation of Leiba.*

Even if, for the sake of argument, Leiba could be modified as suggested by the Examiner, this is not enough to make the Examiner’s proposed modification obvious. [MPEP 2143.01; see also MPEP 2142.01 and 2145.X.C and D.] In fact, due to the nature of the conformance testing in Leiba, in that there is not a software specification to compile into an intermediate language, the Examiner’s proposed modification changes the principle of operation of Leiba and is thus improper. [See In re Ratti, 270 F.2d 810, MPEP § 2143.01.] In addition, Leiba and Willia teach away from the combination suggested by the Examiner.

Leiba is directed toward conformance testing servers where test commands (taken from supplied test data) passed to a server generate a series of responses from the server. The responses can be categorized, such categories including mandatory responses, optimal responses, forbidden responses, etc. It is not directed toward generating test cases from a language specification, and, as previously discussed, does not itself make use of a language specification. Servers, specifically, are conformance-tested in Leiba, and as such, “the server conformance tester provides a mechanism for testing multi-way coordination among client sessions...[where]...clients coordinate their activities with the server in a synchronized fashion.” [Leiba, 4: 10-15.] In Willis, test cases are generated and presented to a “tool under test.” [Willis 3: 18-24, FIG. 1.] There is no suggestion in Willis that the tool-under test procedure can be used to substitute for the specific requirements of the client-server related tests, and as such, Willis teaches away from Leiba.

Furthermore, the test commands in Leiba used to test the Server are input from “Test Data” which is described as having two parts: a test configuration file and a test input file. [Leiba 4: 56-65]. Conversely, the test commands (called “test cases”) in Willis are generated and classified from a “manually generated, formal specification of the language being generated.” [Willis, 2:57-58.] There is no indication, suggestion, or teaching that the test cases generated in Willis are interchangeable with the Test Data in Leiba. As such, Willis teaches away from Leiba.

Moreover, the expected responses in Leiba “may include a plurality of responses, each response including at least one component and the method may further include the steps of permitting non-deterministic ordering of the responses and permitting ordering of components within each response. [Leiba, 2:26-32.] The tester in Leiba also “permits the expected responses to be marked as ‘mandatory’, ‘optional’, or ‘forbidden’.” [Leiba, 4:31-33.] The test cases in Willis to be useful in Leiba must also be able to generate response data which allows the same values as those in Leiba. However, Willis, again, makes no mention of ordering components within a response; makes no mention of permitting non-deterministic ordering, etc. Therefore, Willis teaches away from Leiba, as the types of responses that each generate are different.

In addition, Lieba has no mention of a language specification, a test case compiler, a test case interpreter, or a test case generation environment, all found in Willis.

***B. Lieba and Willis, taken separately or in combination, fail to teach or suggest at least one limitation of claim 1.***

Amended Claim 1 now reads:

A computer-implemented method of conformance-testing a software implementation with a software specification which defines proper behavior of the implementation, the method comprising:

applying software implementation source code and at least a portion of the software specification to produce a conformance-test enabled implementation comprising portions of the software implementation and portions of the software specification integrated into a same body of code wherein nondeterministic choices of the software specification result in assigning a corresponding choice of the conformance-test enabled implementation to a variable, wherein at least one procedure comprises at least one portion of the software implementation and at least one portion of the software specification; and the conformance-test enabled implementation comprising a test that the variable comprises one of the nondeterministic choices of the software specification;

compiling the software implementation source code from a first high-level language into an intermediate language;

compiling the software specification from a second high-level language into the intermediate language; and

producing the conformance-test enabled implementation in the intermediate language. [Emphasis added.]

Leiba and Willis, either together or in combination fail to teach or suggest “applying software implementation source code and at least a portion of the software specification to produce a conformance-test enabled implementation comprising portions of the software implementation and portions of the software specification integrated into a same body of code” as recited in amended Claim 1.

Applicants have further pointed out the nature of the software specification by adding the language below to claim 1:

A computer-implemented method of conformance-testing a software implementation with a software specification which defines proper behavior of the implementation,

Support for this change is found in the specification at page 6 line 7 to page 7, line 4.

In Leiba, a server is conformance tested. The server has requirements that are being tested; such requirements are specified in an RFC document. [See Leiba 1:28-37.] The server being tested in

Leiba has its runtime behavior compared to specified behavior as defined by test data. [See Leiba, FIG. 1, and sec. 1A, above.]

The server being tested is fed test commands from a test engine. The server responds, with the responses given by the server then fed into a response analyzer. [See Sec. 1A, above.] The test commands are generated from black box test data 100 [FIG. 1] which itself consists of a test input file (TIF) and a test configuration file. [Leiba 4:55-65.] The TIF includes “specific protocol commands and all the necessary control elements to manage the flow of the test process. The control elements may include local and global variables, grouping blocks, delays, serialization, iterations, and decision controls.” [Leiba, 4:61-65.] A test generator may use the information in the TIF and the test configuration file to generate various information to be fed into the server to be tested. [Leiba 3:1-24.] However, the relationship between the test generator, the TIF, the test configuration file, and the requirements, which define the rules of the base protocol, is never mentioned, let alone taught or suggested. The patent is silent on how the requirements for server behavior are turned into test data. “The software specification” is not discussed, as required in claim 1, let alone “using a portion of the software specification” … “to produce a conformance-test enabled implementation.”

The server itself acts as a black box, receiving commands from the test engine, processing them, and passing the responses back to the response analyzer. “Software implementation source code” is not discussed with reference to the server. A software implementation is defined in the specification as follows: “A software implementation’s runtime behavior may be compared against the specified behavior to identify non-conformities. This process is referred to as conformance testing.” [Specification, p. 2, ll. 20-22.] As the server in Leiba is being tested, if a software implementation were being used, it would be expected to be the Server’s software implementation. However, there is no discussion of a software implementation of the server, let alone “applying software implementation source code and at least a portion of the software specification to produce a conformance-test enabled implementation comprising portions of the software implementation and portions of the software specification.”

Moreover, as there is no mention of a software specification and no mention of a software implementation, there is, likewise, no mention of “applying software implementation source code and at least a portion of the software specification to produce a conformance-test enabled implementation comprising portions of the software implementation and portions of the software specification

integrated into a same body of code" as required by the claims.

Leiba does state "The response analyzer may include a test generator component for generating multiple request sessions to the server application according to an input specification." [Leiba 3:3-6.] However, the "input specification" of Leiba is not a "software specification *which defines proper behavior of the implementation*" as required by claim 1. The input specification is used to "generate multiple request sessions;" that is, it defines how to stress test the server by bombarding it with requests. There is no indication that input specification is in any way related to the communication protocol being tested.

Willis also fails to teach or suggest the above-cited language of claim 1. Since the cited references fails to describe at least one element recited in claim 1, Applicants believe the claim as amended is not subject to a 103 rejection and request the rejection be withdrawn. Therefore, for these reasons also, claim 1 is now allowable.

#### ***C. Claims 2-6***

Claims 2-6 depend from claim 1. Since they depend from claim 1, they should be allowed for at least the reasons stated for claim 1. In view of the foregoing discussion of claim 1, the merits of the separate patentability of dependent claims 2-6 are not belabored at this time. Claims 2-6 should be allowable. Such action is respectfully requested.

#### ***D. Claims 12-14, 16-17***

Amended independent claim 12 recites the following language:

12. An article comprising:

a machine-readable storage medium comprising instructions to generate a conformance-test enabled implementation of a software specification, the instructions, when executed by a computer system, resulting in:

applying a software implementation source code and at least a portion of the software specification to produce the conformance-test enabled implementation comprising a same body of code with portions from both the software implementation and the software specification in the same procedure, wherein nondeterministic choices of the software specification result in assigning a corresponding choice of the conformance-test enabled implementation to a variable; and the conformance-test enabled implementation comprising a test that the variable comprises one of the nondeterministic choices of the software specification;

compiling the software implementation from a first high-level language into an intermediate language;

compiling the software specification from a second high-level language into the intermediate language; and

producing the conformance-test enabled implementation in the intermediate language. [Emphasis Added.]

In light of the discussion in sections 2A and 2B discussing claim 1, claim 12 is also patentable, in similar grounds, which will not be belabored here. Claims 13-14, and 16-17 depend from claim 12 and include all of the language of claim 12. As noted in section 2A, Leiba fails to teach or suggest language of claim 12. Willis also fails to teach or suggest the above-cited language of claim 12. Therefore, claim 12 is now allowable.

Claims 13-14, and 16-17 depend from claim 12 and include all of the language of claim 12. As they depend from an allowable claim, they are themselves allowable. Such action is respectfully requested.

#### *E. Claims 23-25, 27-28*

Amended claim 23 reads:

An apparatus comprising:

a processor; and

a machine-readable medium comprising instructions to generate a conformance-test enabled implementation of a software specification, the instructions, when executed by the processor, resulting in:

applying a software implementation and the software specification to produce the conformance-test enabled implementation a same body of code with portions from both the software implementation and the software specification integrated into at least one procedure, wherein nondeterministic choices of the software specification result in assigning a corresponding choice of the conformance-test enabled implementation to a variable; and the conformance-test enabled implementation comprising a test that the variable comprises one of the nondeterministic choices of the software specification;

compiling the software implementation from a first high-level language into an intermediate language;

compiling the software specification from a second high-level language into the intermediate language; and

producing the conformance-test enabled implementation in the intermediate language.

In light of the discussion in sections 2A and 2B discussing claim 1, claim 23 is also patentable,

in similar grounds, which will not be belabored here. Willis also fails to teach or suggest the above-cited language of claim 23. Therefore, claim 23 is now allowable. Claims 24-25 and 27-28 depend from claim 23 and include all of the language of claim 23. As they depend from an allowable claim, they are themselves allowable. Such action is respectfully requested.

### 3. *Patentability over Leiba*

The Office has asserted a rejection of claims 7-11, 18-22, 27, and 28-32 under 35 USC §102(e) as anticipated by Leiba, U.S. Patent No. 5,883,661 (“Leiba”). Applicants respectfully traverse. For a 102(e) rejection to be proper, the cited art must show each and every element as set forth in a claim. (See MPEP § 2131.01.) However, the cited art does not so show. For example, with respect to claim 7, Leiba does not teach or suggest a method for “an identification of a mandatory call . . . wherein the mandatory call comprises a call to at least one method in at least one class different than the class of the software object.”

#### *A. Claim 7*

Applicants respectfully submit that Leiba fails to anticipate amended claim 7, because Leiba fails to teach or suggest “the software object comprising at least one instruction which, when executed by a computer system, causes an identification of a mandatory call comprised by the software specification to be stored in a memory of the computer system; *wherein the mandatory call comprises a call to at least one method in at least one class different than the class of the software object.*” Specifically, amended claim 7 recites:

7. A computer-implemented method of conformance-testing a software implementation with a software specification, the method comprising:  
producing a software object organized such that a step of the software specification is surrounded by a corresponding code section of the software implementation in the software object, the software object having a class; and  
the software object comprising at least one instruction which, when executed by a computer system, causes an identification of a mandatory call comprised by the software specification to be stored in a memory of the computer system;  
wherein the mandatory call comprises a call to at least one method in at least one class different than the class of the software object. [Emphasis added.]

The new language is supported in the specification, as shown below:

If *calls to methods in other classes* (henceforth ‘mandatory calls’) are indicated in the specification, a check is made to confirm that those calls are made by the implementation. A check is made during each mandatory call to confirm that the state of the implementation remains in conformance with the specification.  
[Application, Page 20, lines 19-23.][Emphasis added.]

Leiba fails to show the feature of “mandatory call, wherein the mandatory call comprises a call to at least one method in at least one class different than the class of the software object.” Classes are not mentioned in Leiba, let alone a “mandatory call, wherein the mandatory call comprises a call to at least one method in at least one class different than the class of the software object.”

At least for this reason, Leiba fails to teach or suggest at least one element of claim 7. Claim 7 is thus in condition for allowance.

#### ***B. Claims 8-11***

Claims 8-11 depend from claim 7. Since they depend from an allowable claims, they should be allowed for at least the reasons stated for claim 7. In view of the foregoing discussion of claim 7, the merits of the separate patentability of dependent claims 8-11 are not belabored at this time. Claims 8-11 should be allowable. Such action is respectfully requested.

#### ***C. Claims 18-22, 29-31***

Amended independent claim 18 recites the following language:

18. An article comprising:  
a machine-readable storage medium comprising instructions to generate a conformance-test enabled implementation of a software specification, the instructions, when executed by a computer system, resulting in:  
producing a software object organized such that a step of the software specification is surrounded by a corresponding code section of the software implementation in the software object, the software object having a class; and  
the software object comprising at least one instruction which, when executed by a computer system, causes an identification of a mandatory call comprised by the software specification to be stored in a memory of the computer system; wherein the mandatory call comprises a call to at least one method in at least one class different than the class of the software object. [emphasis added.]

Amended Independent claim 29 recites the following language:

29. An apparatus comprising:

a processor; and

a machine-readable medium comprising instructions to generate a conformance-test enabled implementation of a software specification, the instructions, when executed by a computer system, resulting in:

producing a software object organized such that a series of steps of the software specification and a corresponding code section of the software implementation are enmeshed in the software object, the software object having a class;

the software object comprising at least one instruction which, when executed by a computer system, causes an identification of a mandatory call comprised by the software specification to be stored in a memory of the computer system, wherein the mandatory call comprises a call to at least one method in at least one class different than the class of the software object; and

the software object comprising at least one instruction which, when executed by the computer system, causes a test that the state of the conformance-test enabled implementation conforms to the software specification during the mandatory call.  
[emphasis added.]

These claims both recite language related to the feature of a “mandatory call which *comprises a call to at least one method in at least one class different than the class of the software object*” which is not present in Leiba, as described above, with reference to the discussion concerning claim 7. In view of the foregoing discussion of claim 7, the merits of the separate patentability of claims 18 and 29 are not belabored at this time. Claims 18 and 29 should be allowable. Such action is respectfully requested. Furthermore, claims 19-22 depend from claim 18, and as the dependent claims of an allowable claim, should also be allowable. Similarly, claims 30-31 depend from claim 29, and as the dependent claims of an allowable claim, should be allowable themselves. Such action is respectfully requested.

#### **4. Request For Interview**

If any issues remain in light of these remarks and amendments, the Examiner is formally requested to contact the undersigned attorney to arrange a telephonic interview. It is believed that a brief discussion of the merits of the present application may expedite prosecution. Applicants submit the preceding formal Amendment and the above remarks so that the Examiner may fully evaluate Applicants’ position, thereby enabling the interview to be more focused.

This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

**Conclusion**

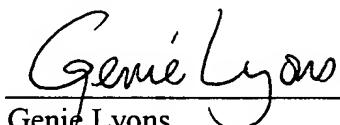
The claims in their present form should now be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600  
121 S.W. Salmon Street  
Portland, Oregon 97204  
Telephone: (503) 595-5300  
Facsimile: (503) 595-5301

By

  
\_\_\_\_\_  
Genie Lyons  
Registration No. 43,841